2 Biclustering

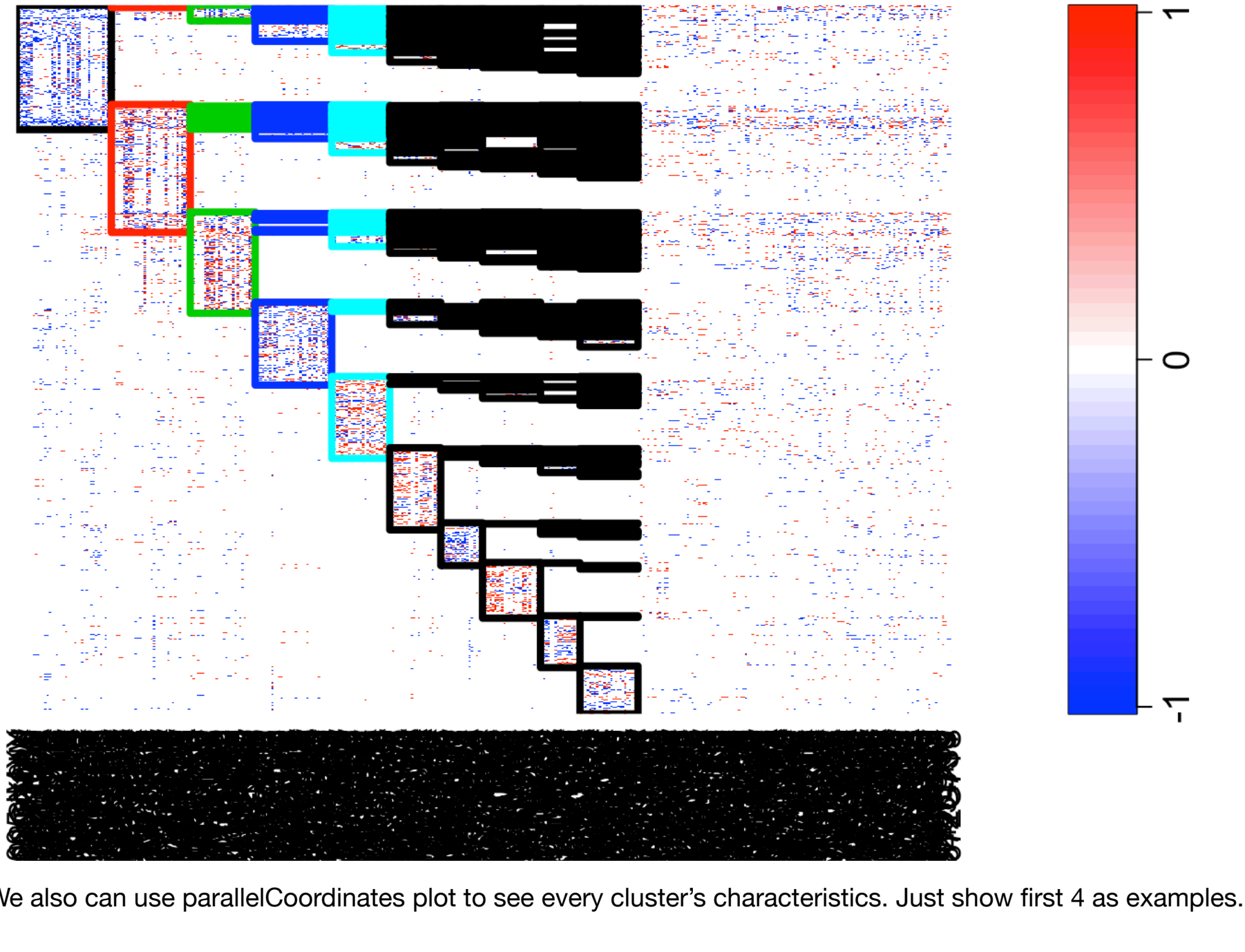
Biclustering is an important new technique in two way data analysis. With biclustering, we can do simultaneous clustering of 2 dimensions; Large datasets where clustering leads to diffuse results; Only parts of the data influence each other.

2.1 Biclustering based on user id and product id:

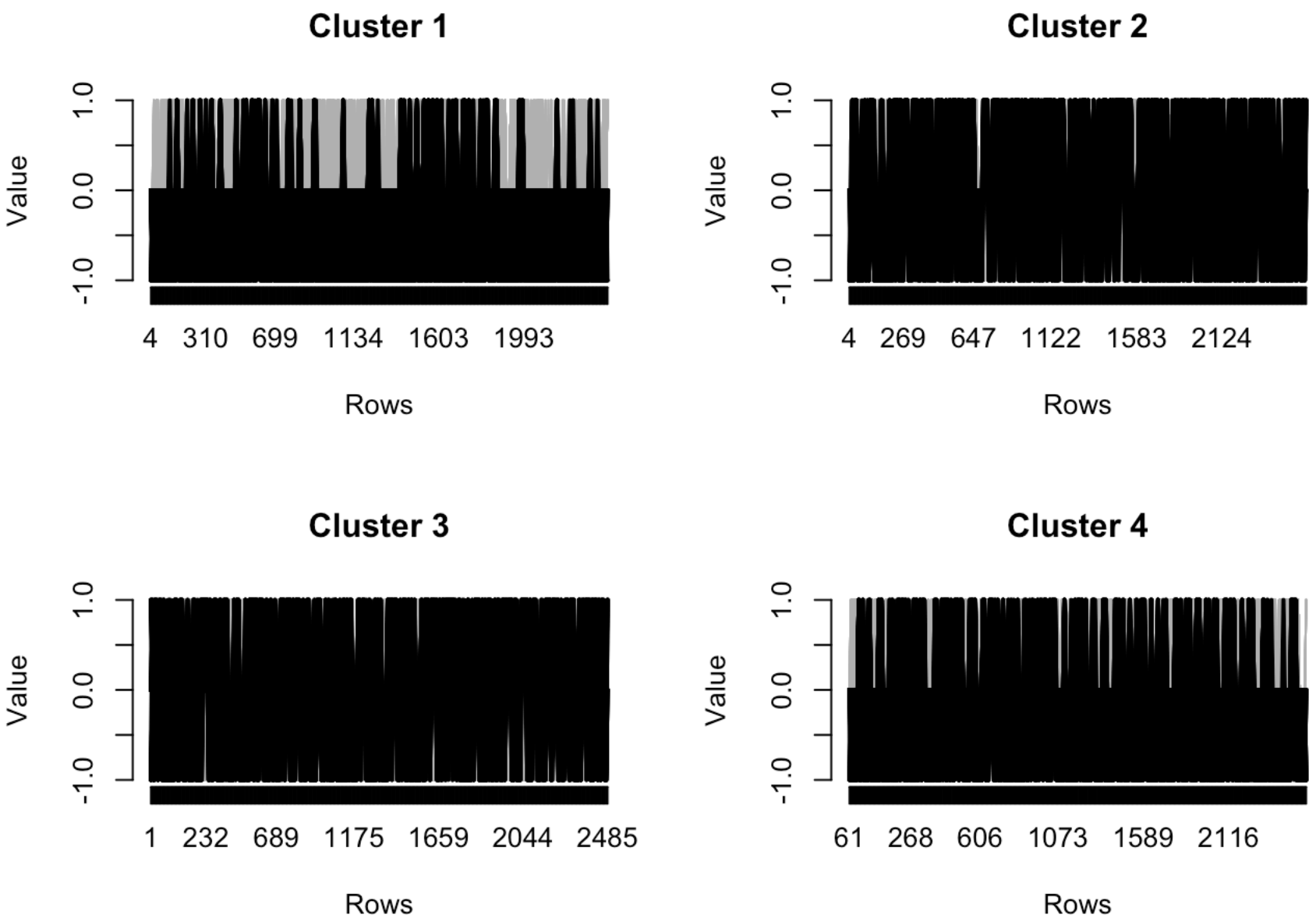
First of all, we do biclustering based on user id and product id. Due to concerns about memory and running time, we extract a subset from the original dataset. For movies (403), those that are reviewed by experts and have more than 35 reviews are selected. For user id (2486), only reviewers who have watched the above movies are chosen.

To creating the sparse matrix, user id represents rows, and product id represents columns. If user i gives review score 5 to product j, then the entry (i,j) would be 1, if user i gives review score below 5 to product j, then the entry (i,j) would be -1, if user i hasn't given a review to product j, then the entry (i,j) would be 0.

```
## Bicluster 1........
## Bicluster 2..........
## Bicluster 3........
## Bicluster 4.....................................
## Bicluster 5.........................
## Bicluster 6..............
## Bicluster 7...................
## Bicluster 8...............
## Bicluster 9......................
## Bicluster 10.............
```



We also can use parallelCoordinates plot to see every cluster’s characteristics. Just show first 4 as examples.



So far, we can give a recommendation system according to this approach. For example, if user “A17IW44FV0HUTY” wants to find some recommendations, we will firstly locate his/her cluster, second we select all movies that are in this cluster and haven’t been watched by user “A17IW44FV0HUTY”, and then we recommend top n (n is a customized number of recommendations, for instance, n=3) movies based on average review socre.

Finally, we have the results.

```
## [1] "Jaws"
## [2] "United 93 [DVD + Digital Copy] (Universal's 100th Anniversary)"
## [3] "Million Dollar Baby"
```

On the other hand, if one user likes movie “B00000I4XR”, which is “Jaws”, again, we will firstly locate its cluster, second we select all other movies that are in this cluster, and then we recommend top n (n is a customized number of recommendations, for instance, n=3) movies based on average review socre.

We now have the results.

```
recommend_new_final[1:3]
```

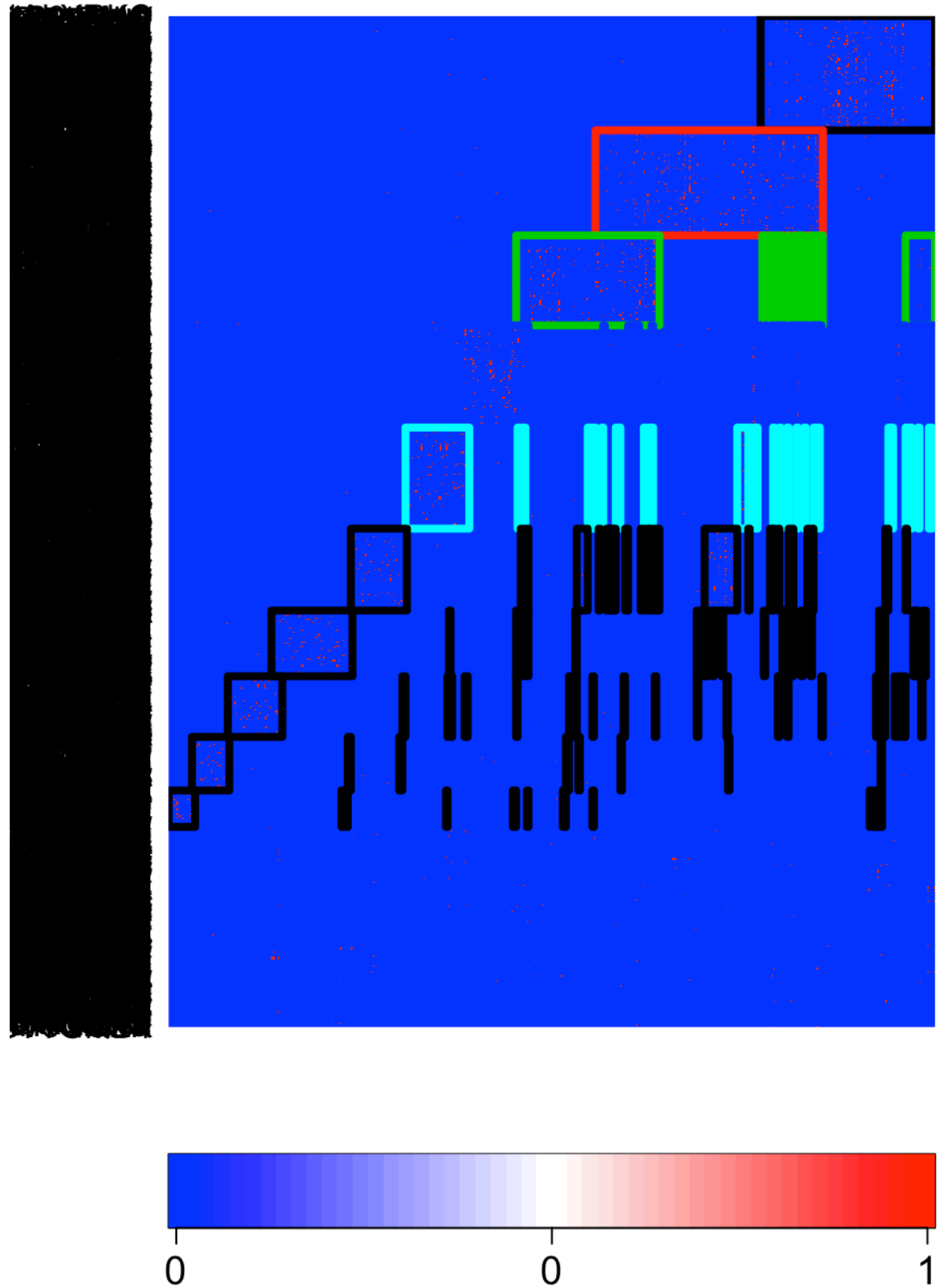
```
## [1] "United 93 [DVD + Digital Copy] (Universal's 100th Anniversary)"
## [2] "Master and Commander - The Far Side of the World [VHS]"
## [3] "Pan's Labyrinth (New Line Two-Disc Platinum Series)"
```

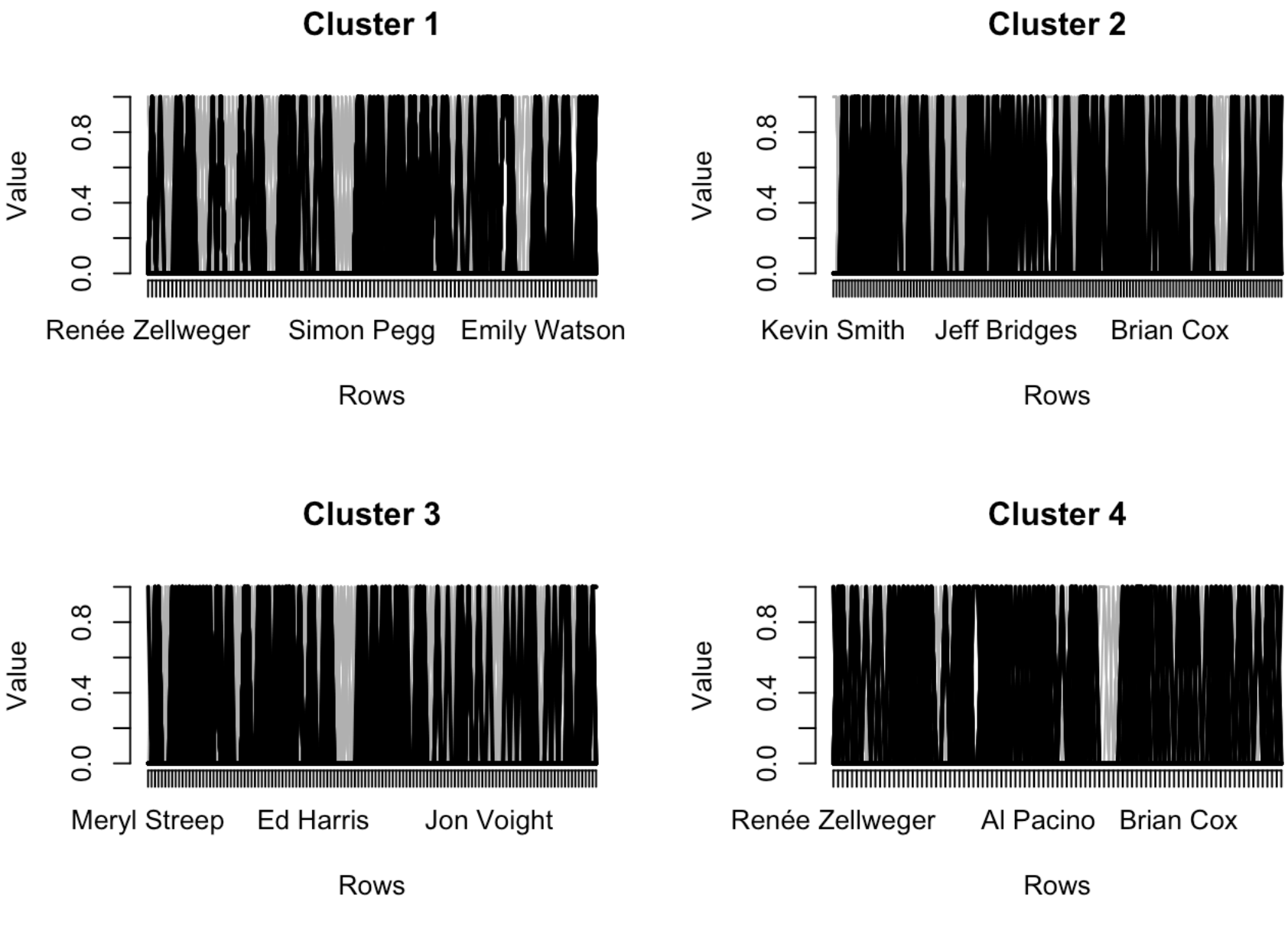
2.2 Biclustering based on actors' and directors' names and product id:

Second, we do biclustering based on actors and directors and product id. Also, since the dataset is too large, we extract a subset with last 1000 movies and first 1000 actors from the original dataset.

To creating the sparse matrix, actors' names represent rows, and product id represents columns. If actor i performed in movie j, then the entry (i,j) would be 1, otherwise would be 0.

```
## Bicluster 1....
## Bicluster 2..............
## Bicluster 3.....
## Bicluster 4....
## Bicluster 5....
## Bicluster 6....
## Bicluster 7....
## Bicluster 8....
## Bicluster 9....
## Bicluster 10....
```


0
0
1



So far, we can give a recommendation system according to this approach. For example, if one user inputs a movie that he/she likes. Again, we will firstly locate its cluster, second we select all movies that are in this cluster, and then we recommend top n (n is a customized number of recommendations, for instance, n=3) movies based on average review socre.

For example, if the input movie id is “B0093ICOE0”, which is “Watchmen Collector’s Edition: Ultimate Cut + Graphic Novel [Blu-ray]”, then let’s see what we’ll get.

Finally, we have the results.

```
## [1] "Iron Man - Spanish Version"
## [2] "Eternal Sunshine Of The Spotless Mind (Eterno Resplandor De Una Mente Sin Rec
uerdos) [NTSC/REGION 1 & 4 DVD.Import-Latin America]"
## [3] "Collateral [Blu-ray]"
```

2.3 Word cloud of popular actors and directors

Wordcloud of Directors

Wordcloud of Actors

2.4 Recomendation Algorithm 2 & 3:

This recommendation algorithm based on biclustering. Intuitively, the results should share some similarities with the input (They are in the same cluster).

Advantage: 1. do simultaneous clustering of 2 dimensions 2. large datasets where clustering leads to diffuse results 3. only parts of the data influence each other

Disadvantage: 1. slow, but not the slowest one 2. not stable, need to tune parameters of the model